СИММЕТРИИ АЛГОРИТМОВ МАТРИЧНОГО УМНОЖЕНИЯ

В.П. Бурименко

Лаборатория теории конечных групп Института математики НАН Беларуси,
Кирова 32а, 246000 Гомель, Беларусь
vpburich@gmail.com

Данная работа связана с проблемой быстрого умножения матриц. Рассматриваются некоммутативные (в смысле [1]) алгоритмы умножения матриц. Пусть K — поле, R — ассоциативная K-алгебра, X и Y — $m \times n$ и $n \times p$ матрицы над R. Вычисление произведения XY обычным способом ("строка на столбец") требует mnp умножений в R. Однако, существуют более быстрые алгоритмы. При $m = n = p = 2$ достаточно 7 умножений (алгоритм Штрассена, [2]), при $(m, n, p) = (2, 3, 3)$ — 15 умножений (алгоритм Хопкрофта, [3]), $m = n = p = 3$ — 23 умножения (алгоритм Ладермана, [4]), при $m = n = p = 2l$ — $(n^3 - 4n)/3 + 6n^2$ умножений (алгоритм трилинейного агрегирования Пана, [5]). Обозначим минимальное необходимое число умножений через $r(m, n, p)$ (вообще говоря, $r(m, n, p)$ зависит от K).

Если есть (нетривиальный) алгоритм умножения $m \times n$ матрицы на $n \times p$ матрицу, требующий r умножений, его можно применить рекурсивно и показать, что умножение двух квадратных $N \times N$ матриц над K требует $O(N^\tau)$ арифметических операций, где $\tau = 3 \log_{mnp} r$ (см. [1]). Поэтому нахождение верхней оценки для $r(m, n, p)$ при конкретных m, n, p — практически (и теоретически) важная задача.

Алгоритмы матричного умножения тесно связаны с разложениями тензоров. Пусть $\widetilde{U} = U_1 \otimes \ldots \otimes U_l$ — тензорное произведение нескольких пространств над K. Тензор $u \in \widetilde{U}$ называется *разложимым*, если $u = u_1 \otimes \ldots \otimes u_l$, $u_i \in U_i$. Далее, если $t \in \widetilde{U}$ — произвольный тензор, и $\mathcal{A} = \{t_1, \ldots, t_s\}$ — множество разложимых тензоров такое, что $t_1 + \ldots + t_s = t$, тогда $\mathcal{A}$ называется *алгоритмом* (длины s) *для вычисления тензора* t. Минимальная длина $s = |\mathcal{A}|$ называется *рангом* тензора t, и обозначается через $\mathrm{rk}(t)$.

Через $M_{ab} = M_{a,b}(K)$ обозначим пространство $a \times b$ матриц над K. Для данных m, n, p положим $L_1 = M_{mn}$, $L_2 = M_{np}$, $L_3 = M_{pm}$, $L = L_1 \otimes L_2 \otimes L_3$, и рассмотрим тензор

$$\langle m, n, p \rangle = \sum_{1 \le i \le m,\ 1 \le j \le n,\ 1 \le k \le p} e_{ij} \otimes e_{jk} \otimes e_{ki} \in L.$$

Хорошо известно, что алгоритмы, вычисляющие произведение $m \times n$ и $n \times p$ матриц, находятся в биекции с алгоритмами, вычисляющими тензор $\langle m, n, p \rangle$, и что

$$r(m, n, p) = \mathrm{rk}(\langle m, n, p \rangle).$$

Таким образом, изучение алгоритмов матричного умножения — это в точности изучение разложений тензоров $\langle m, n, p \rangle$.

Автор считает, что одним из плодотворных путей для построения экономичных алгоритмов (т.е., коротких разложений тензоров) является исследование алгоритмов, обладающих нетривиальной группой симметрии. Дадим необходимые определения.

Пусть $\widetilde{U} = U_1 \otimes \ldots \otimes U_l$ — тензорное произведение, как выше. Автоморфизм $g \in GL(\widetilde{U})$ *разложим*, если он согласован, в очевидном смысле, со структурой тензорного произведения на $\widetilde{U}$ (при этом g может переставлять факторы $U_1, \ldots, U_l$ нетривиальным образом). Группу всех разложимых автоморфизмов обозначим $S(\widetilde{U}) = S(U_1, \ldots, U_l)$. Для данного тензора $t \in \widetilde{U}$ определим его *группу изотропии* $\Gamma(t)$ как

$$\Gamma(t) = \{g \in S(\widetilde{U}) \mid g(t) = t\}.$$

Далее, для данного алгоритма $\mathcal{A} = \{t_1, \ldots, t_s\}$, вычисляющего t, определим его *группу автоморфизмов*

$$\mathrm{Aut}(\mathcal{A}) = \{g \in S(\widetilde{U}) \mid g(\mathcal{A}) = \mathcal{A}\}.$$

Ясно, что всегда $\mathrm{Aut}(\mathcal{A}) \leq \Gamma(t)$.

Очевидно, первый шаг при исследовании алгоритмов с точки зрения их симметрии — определить группы автоморфизмов для известных хороших алгоритмов. Это является основным результатом настоящей работы.

Пусть $\mathcal{S}$, $\mathcal{H}$, $\mathcal{L}$, $\mathcal{P}_{2l}$ означают алгоритмы Штрассена, Хопкрофта, Ладермана и Пана, соответственно (точнее, соответствующие алгоритмы, вычисляющие тензоры $\langle 2,2,2\rangle$, $\langle 2,3,3\rangle$, $\langle 3,3,3\rangle$, $\langle 2l,2l,2l\rangle$). Доказана следующая теорема.

Теорема 1. *Имеют место изоморфизмы*

$$\mathrm{Aut}(\mathcal{S}) \cong S_3 \times S_3\,, \quad \mathrm{Aut}(\mathcal{H}) \cong S_3 \times Z_2\,,$$

$$\mathrm{Aut}(\mathcal{L}) \cong S_4\,, \quad \mathrm{Aut}(\mathcal{P}_{2l}) \cong S_l \times Z_2 \times S_3\,.$$

(Конечно, указанные группы автоморфизмов найдены в явном виде, а не только с точностью до изоморфизма.)

В ходе исследования найдена группа изотропии $\Gamma(t)$, где $t = \langle m,n,p\rangle$, для любых m, n, p.

Теорема 2. *Пусть $\Gamma(t)$ — группа изотропии тензора $t = \langle m,n,p\rangle$, и $\Gamma^0(t)$ — подгруппа элементов $g \in \Gamma(t)$, сохраняющих факторы произведения $L_1 \otimes L_2 \otimes L_3$ (т.е. вида $g = g_1 \otimes g_2 \otimes g_3$, $g_i \in GL(L_i)$). Тогда $\Gamma^0(t)$ совпадает с группой всех преобразований вида*

$$T(a,b,c) : x_1 \otimes x_2 \otimes x_3 \mapsto ax_1b^{-1} \otimes bx_2c^{-1} \otimes cx_3a^{-1}\,,$$

где $x_i \in L_i$, $a \in GL_m(K)$, $b \in GL_n(K)$, $c \in GL_p(K)$. Факторгруппа $\Gamma(t)/\Gamma^0(t)$ изоморфна одной из групп 1, Z_2 или S_3.

Конечно, гораздо более важной, чем теорема 1, является следующая

Обратная задача. Для данной подгруппы $G \leq \Gamma(t)$ и $r \geq 1$ описать все G-инвариантные алгоритмы длины r, вычисляющие t.

(Именно в ходе решения задач такого вида и возможно было бы найти новые алгоритмы).

Литература

1. Bürgisser P., Clausen M., Shokrollahi M. A. *Algebraic Complexity Theory.* Springer, 1997.

2. Strassen V. *Gaussian elimination is not optimal* // Numer. Math. 1969. V. 13. № 4. P. 354–356.

3. Hopcroft J. E., Kerr L. R. *On minimizing the number of multiplications necessary for matrix multiplication* // SIAM J. Appl. Math. 1971. V. 20. P. 30–36.

4. Laderman J. *A noncommutative algorithm for multiplying 3×3 matrices using 23 multiplications* // Bull. Amer. Math. Soc. 1976. V. 82. P. 180–182.

5. Pan V. Ya. *Strassen algorithm is not optimal. Trilinear technique of aggregating, uniting and cancelling for constructing fast algorithms for matrix multiplication* // Proc. 19th Annual conference on Foundations of Computer Science, Ann Arbor, 1979; pp. 166–176.